Extracting Features from 3D Unstructured Meshes for Interactive Visualization

Kwan-Liu Ma and Victoria Interrante
Institute for Computer Applications in Science and Engineering

Abstract

This paper describes techniques, based on the extraction of geometric features, for facilitating the visualization and interactive manipulation of the typically very large and dense three-dimensional unstructured grids used in aerodynamics calculations.

We discuss the difficulties that scientists currently face in efficiently and effectively displaying these meshes and propose methods for using geometric feature lines to clearly and concisely indicate the essential structural detail of the model while eliminating much of the unnecessary visual clutter.

We describe the perceptual importance of specific viewpoint-dependent and view-independent features, discuss the practical implementation of simple but effective algorithms for identifying these features (taking into consideration both local and global criteria), and demonstrate the performance of each proposed technique on various types of data sets.

†This research was supported by the National Aeronautics and Space Administration under NASA contract NAS1-19480 while the author was in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681-0001.

1 Introduction

The three-dimensional unstructured grids used for the numerical simulation of 3D flow are generally very large in size and irregular in both shape and resolution. Even the simplest renderings of many of these meshes can be time-consuming to compute on an average desktop workstation, and once an image is produced it can be difficult to adequately perceive relevant geometric structure through the tangle of overlapping lines. Figure 1 gives an example of a typical rendering of a CFD grid. This picture illustrates a surface mesh, consisting of 127,544 triangles, that was extracted from an unstructured volume grid containing 4,607,585 tetrahedra across 804,056 points. Figures 2-4 illustrate some of the standard techniques that are commonly used in practice to decrease the rendering time required for the display of such datasets. (These techniques include rendering only the gridpoints of the surface mesh, skipping every nth element, or substituting a coarser grid.)

By directly identifying and extracting a small set of perceptually significant geometric features from the surface mesh and displaying these in place of the full model we may both considerably decrease the rendering latency compared to all of the above-mentioned methods and at the same time improve the comprehensibility of the presented data.

When rendering time is not of critical concern, it can be useful to display feature lines in conjunction with surface or volume-rendered data to highlight essential structural detail of the underlying geometry while preserving the visual prominence of the flow information.

2 Previous Work

The use of feature lines to enhance the communication of geometrical information has a long history in computer graphics. Dooley and Cohen [4] may have been among the first in this field to stress the perceptual importance of silhouette, contour, discontinuity and, in certain cases, isoparametric lines for clarifying geometrical structure in a complex model. They show how various techniques from technical illustration can be used to successfully represent these feature lines in images of CAD models containing multiple overlapping surfaces, but do not discuss how such lines may be identified. Pearson and Robinson [13] show how images resembling artists' line drawings can be computed from two-dimensional digital photographs and they describe how the use of such representations can enable improved bandwidth for visual communication across low speed data lines. Saito and Takahashi [15] propose enhancing shaded renderings of polygonal models with lines representing the locus of first and second order depth discontinuities, and they describe how these lines can be calculated by applying standard gradient operators to a two-dimensional depth map of the scene. Miller [12] demonstrates how tangent sphere accessibility can be used to identify the narrow recesses in the surfaces of polygonally modeled objects that tend, on actual objects,

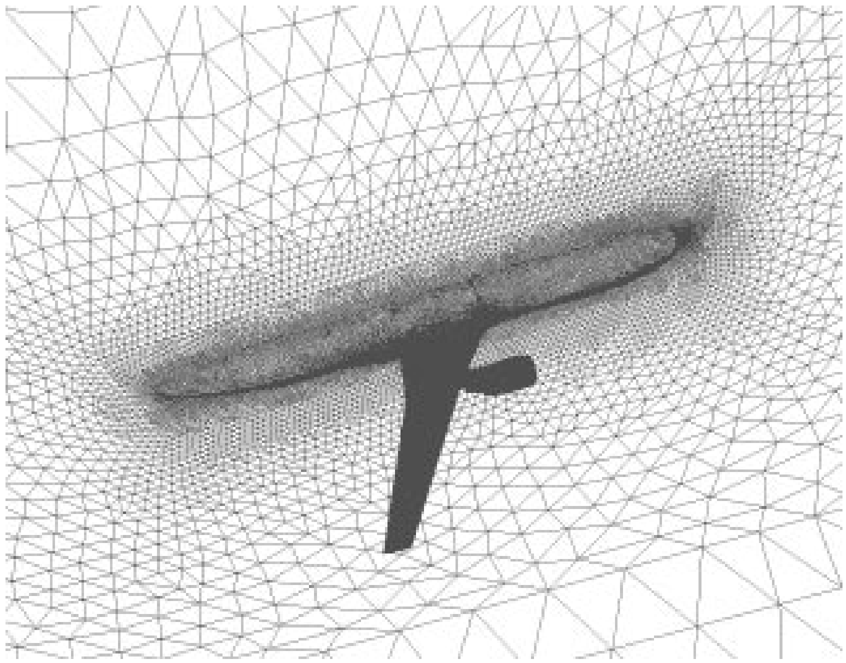

Figure 1: A surface mesh for the numerical simulation of airflow over a low-wing transport aircraft.

Figure 2: The same dataset, with only the gridpoints displayed.

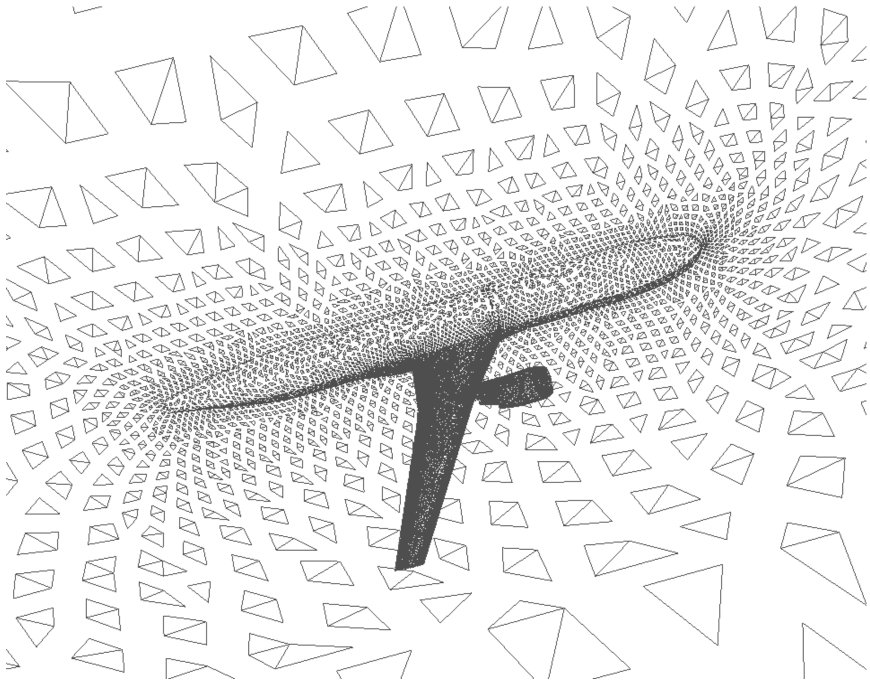

Figure 3: The same dataset, with only every fourth grid element displayed.

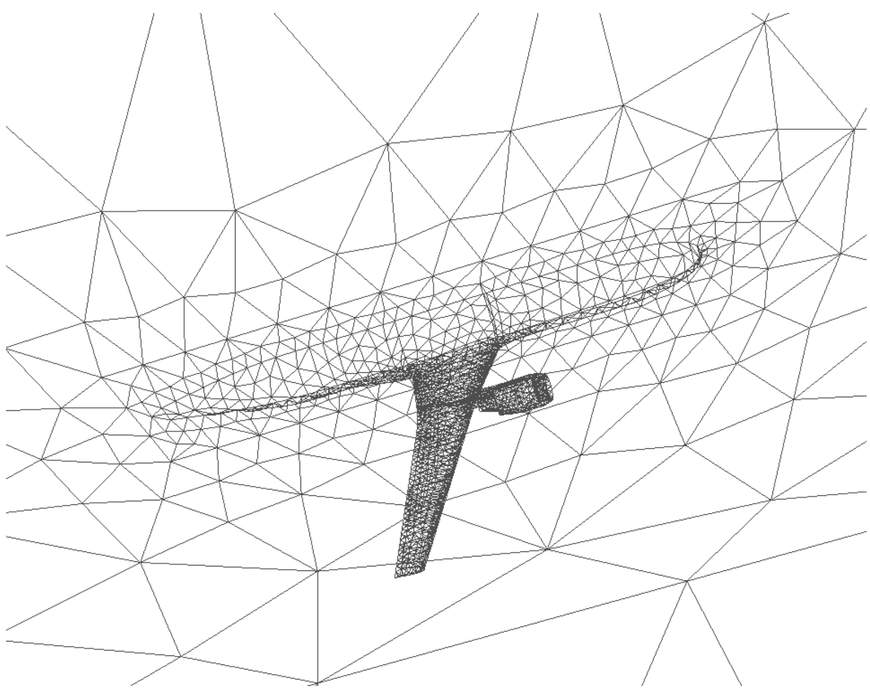

Figure 4: The same surface, modelled by a lower resolution mesh.

to appear darker than would otherwise be expected for a variety of reasons. Interrante, Fuchs and Pizer [8] suggest explicitly marking valley lines (local minima of normal curvature in the local direction of greatest negative curvature) on transparent skin surfaces defined by volume data to allow perceptually relevant surface shape information to be more clearly conveyed without unduly occluding underlying structures.

Mesh simplification represents an alternative to feature line extraction in that it offers the possibility of considerable data reduction while maintaining fidelity to the original model. Although the best of these methods preserve sharp edges either implicitly [3] or explicitly [6], mesh simplification offers only a partial solution to the problem of effectively visualizing dense grids since one still has to contend with the problem of distinguishing the perceptually relevant edges in the rendered image.

3 Feature Lines

Cross-cultural research in pictorial representation [9] indicates that line drawings are a natural and universally understood means of graphical communication, and visual theorists have suggested the possibility of an intrinsic relationship between this type of representation and the way our visual system processes and stores information (e.g. Marr's [11] "primal sketch" theory of visual information processing).

3.1 Silhouettes and Contours

Silhouette and contour curves are the two-dimensional projection of the points on a surface in 3-space where the surface normal is orthogonal to the line of sight [10]. Contour lines mark the depth discontinuities in a two-dimensional image, and silhouette lines separate the figure from the ground. Richards et al. [14] describe the many properties of three-dimensional shape that can be directly inferred from various characteristics of the occluding contour.

Procedures for identifying sihouette edges in polygonal models are very straightforward and have been known for many years. In the case of orthographic projection, one may easily identify silhouette edges in a connected mesh by simply taking the dot product of the transformed view direction with the surface normal of each triangle and looking for differences in sign between adjacent elements. When a perspective projection is used things get slightly more complicated since the viewing direction is not constant over all areas of the scene. However a clear and succinct algorithm for identifying silhouettes under these conditions is given in [7].

Because they are viewpoint-dependent, silhouette and contour curves have to be recomputed continuously as an object is repositioned in space. If the dataset is small, and an orthographic projection is used, it may be possible to perform the necessary calculations

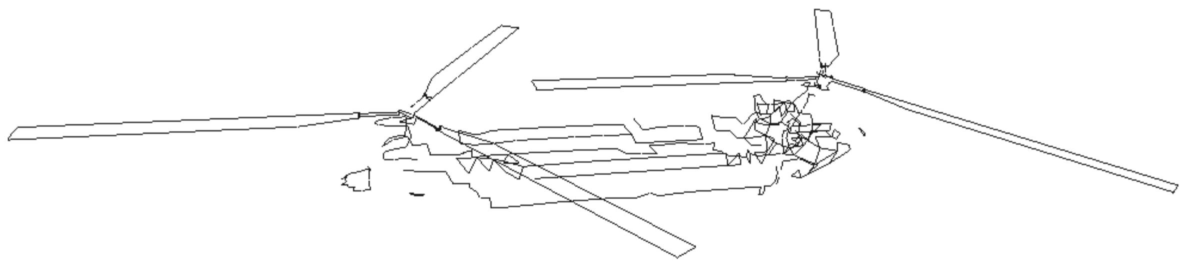

Figure 5: A "quick and dirty" approximation of the silhouette and contour edges of a surface mesh defining a tandem helicopter.

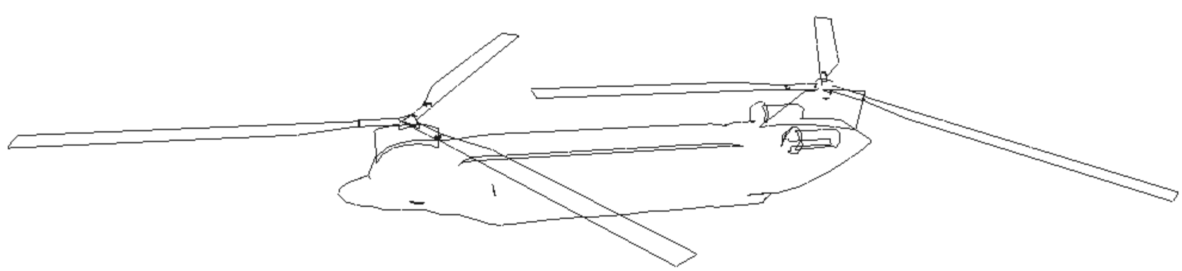

Figure 6: An accurate rendering of the silhouette edges approximated above.

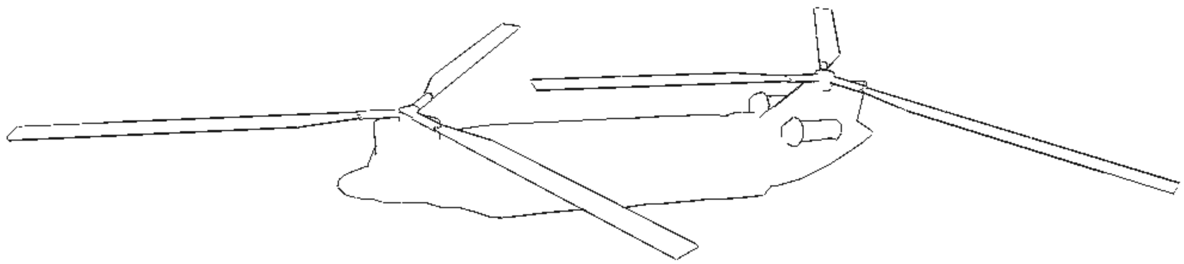

Figure 7: A silhouette line rendering with the hidden lines removed.

quickly enough to permit acceptably responsive update rates, but for large datasets, and particularly under perspective projection, such representations can become prohibitively expensive to recompute on the fly. It is possible, of course, to use a "quick and dirty" estimate of the silhouette edges for more rapid interactive display and then switch to a mode in which silhouettes are more accurately determined once more appropriate viewing parameters have been selected (as an accurate image may take anywhere from 3 to 12 times longer to render than an approximate one), and we have included this features as an option in our system. Figures 5-6 illustrate some of the differences between the two methods.

It can sometimes be useful, for visualization purposes, to graphically emphasize the silhouette and contour curves of an object or to display them in conjunction with shaded renderings of the surface or volume data in situations where rendering latency is not a concern. Using some form of hidden line removal, as illustrated in figure 7, may often improve the clarity of the presentation. Nevertheless, because it is generally faster to just display an edge than it is to determine whether or not the edge is a silhouette (and, especially, to determine whether or not it is occluded), the selective rendering of these features usually does more to solve the problem of image quality than it does to facilitate interaction with the model.

3.2 Ridges and Valleys

A second class of feature lines that are often included in line drawing representations are the lines of intensity discontinuity. These lines are generally viewpoint-independent and correspond to the places on an object where the surface normal changes direction abruptly. On polygonally-defined surfaces such as the numerical calculation grids that we are concerned with here, however, there is a one-to-one correspondence between discontinuity edges and the lines of the mesh. How can we differentiate the few perceptually relevant edges from the many others? There are several considerations that need to be weighed.

The results of psychophysical experiments in shape perception [2] support theories of object recognition [1] based on the subdivision of complex shapes into components along the lines defined by local minima of negative curvature [5]. These are the "valley" lines, and they are important to shape understanding regardless of the sharpness of the curvature across them.

Ridge lines, which are the convex analogue of valleys, appear on the other hand to be perceptually relevant only to the extent that they mark areas of significant curvature discontinuity.

The primary obstacle to extracting ridge and valley lines from faceted objects is the need for a representation of the underlying surface that is sufficiently smooth to allow reasonable estimates of the principal curvatures and principal directions to be obtained.

Since our primary objective for this application is speed, many of the algorithmic avenues that we might otherwise pursue for feature line extraction become computationally impractical and we are faced with the necessity of employing approximate methods to achieve the results that will be adequate for our purposes.

One very simple way to extract a small set of potentially important edges from a connected mesh is to calculate, for each pair of adjacent triangles, the difference between the two surface normal directions and to define as feature lines the edges across which the angle indicated by this quantity exceeds some user-defined global threshold. We have found in practice that this technique actually works remarkably well for the many of the kinds of datasets that we typically deal with. Figure 8 illustrates the results of applying this algorithm to the display of a surface mesh defined, for domain decomposition experiments, over a model of a Commanche helicopter.

There are, of course, many instances in which a method like this will fail to produce satisfactory results. In any locally spherical region for example, where all of the facets will tend to form angles of approximately similar magnitude, we will end up with either every one of these mesh lines being rendered or none of them, depending on the value of the threshold. We also will run into trouble in situations where the magnitude of the curvature across the edges that we would like to show is not easily expressed by a single value but rather varies widely across different areas of the mesh.

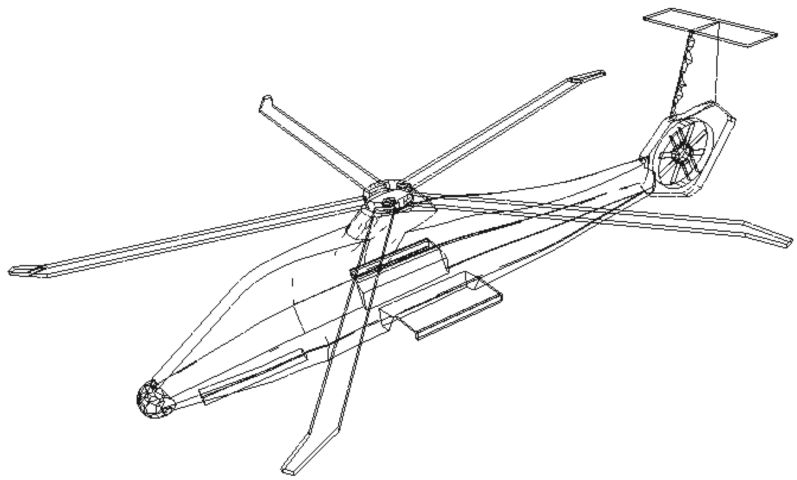

Figure 8: A set of feature lines from a surface mesh over a Commanche helicopter, defined by a global curvature criterion.

An alternative approach to the definition of feature lines begins from the hypothesis that the visual importance of any particular edge can be related to the extent to which the curvature across that edge (approximated by the angle of the normals) is particularly larger than the curvatures in other directions at that same point (indicated by the angles between the normals across the other two edges of the same triangles). Figure 9 illustrates a set of feature lines defined by an algorithm following this procedure.

It is sometimes beneficial to obtain larger scale curvature estimates in a locally smooth region by averaging the normal directions of neighboring triangles. Special precautions need to be taken, if this is done, to avoid averaging across significant discontinuity edges.

4 Directions for Future Work

Neither spherical nor cylindrical structures are especially well described by either of the view-independent feature line representations that we propose. Meshes containing a principal spherical or cylindrical component may be so inadequately represented that whole pieces can be inadvertently left out (on the airplane model, for example, we will see only the wings, engine and tail). Although indications of the missing segments can be restored by complementing the ridge- and valley-like feature lines of section 3.2 with feature lines representing

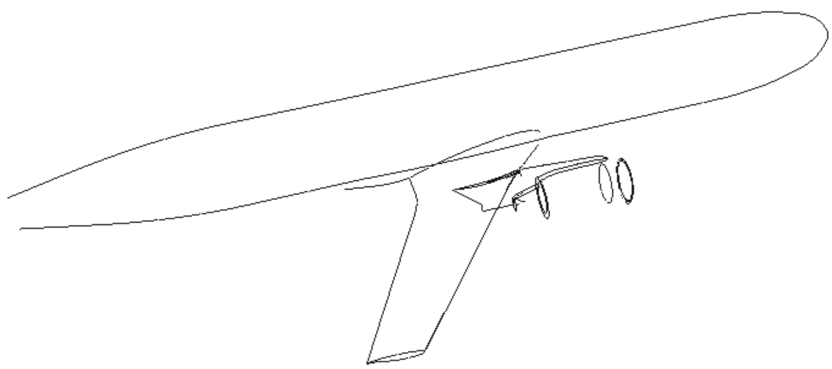

Figure 9: A set of feature lines on the low-wing transport dataset, defined by a local curvature criterion within global limits.

silhouette edges, as shown in figure 10, it would be preferable to come up with a fully viewpoint independent technique that is capable of elegantly portraying these nongeneric structures at unambiguously interactive rates. A representation based on the illustration of object cross-sections locally perpendicular to the direction of the medial axis may show some promise in this regard.

5 Conclusions

We have presented several simple but efficient and effective techniques for extracting perceptually relevant feature lines from unstructured triangular meshes. The methods that we have proposed are very fast and require little or no preprocessing of the data. They allow the clarity of the display to be enhanced while enabling complex models to be interactively manipulated in a fraction of the time that is ordinarily required for a rendering of the complete dataset. We hope that such algorithms can be incorporated into future commercial systems for 3D CFD data display, as we find them very useful and believe that others will too.

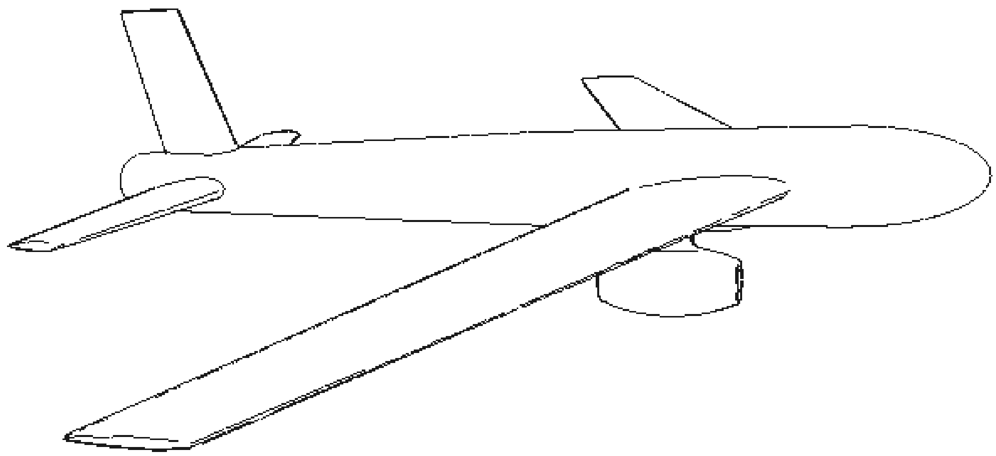

Figure 10: Silhouette and feature lines from a surface mesh over an airplane.

6 Acknowledgments

The grid datasets were provided by Dimitri Mavriplis and Can Ozturan.

References

[1] Biederman, I. Human Image Understanding: Recent Research and a Theory. In *Human and Machine Vision*, A. Rosenfeld, Ed. Academic Press, 1985, pp. 13–57.

[2] Braunstein, M. L., Hoffman, D. D., and Saidpour, A. Parts of Visual Objects: an experimental test of the minima rule. *Perception 18* (1989), 817–826.

[3] Cohen, J., Varshney, A., Manocha, D., Turk, G., Weber, H., Agarwal, P., Brooks, F., and Wright, W. Simplification Envelopes. In *Proceedings of SIGGRAPH '96* (1996), pp. 119–128.

[4] Dooley, D., and Cohen, M. Automatic Illustration of 3D Geometric Models: Lines. In *Proceedings of 1990 Symposium on Interactive 3D Graphics* (1990), pp. 77–82.

[5] Hoffman, D. D., and Richards, W. A. Parts of Recognition. *Cognition 18* (1984), 65–96.

[6] Hoppe, H. Progressive Meshes. In *Proceedings of SIGGRAPH '96* (1996), pp. 99–108.

[7] Hultquist, J. Backface Culling. In *Graphics Gems*, A. S. Glassner, Ed. Academic Press, 1990, pp. 346–347.

[8] Interrante, V., Fuchs, H., and Pizer, S. Enhancing Transparent Skin Surfaces with Ridge and Valley Lines. In *Proceedings of the Visualization '95 Conference* (1995), pp. 52–59.

[9] Kennedy, J. M. *A Psychology of Picture Perception*. Jossey-Bass, 1974.

[10] Koenderink, J. *Solid Shape*. MIT Press, 1990.

[11] Marr, D. Early Processing of Visual Information. *Philosophical Transactions of the Royal Society of London, Series B. Biological Sciences 275* (October 1976), 483–519.

[12] Miller, G. Efficient Algorithms for Local and Global Accessibility Shading. In *Proceedings of SIGGRAPH '94* (1994), pp. 319–326.

[13] Pearson, D. E., and Robinson, J. A. Visual Communication at Very Low Data Rates. *Proceedings of the IEEE*, 4 (April 1985), 795–812.

[14] Richards, W., Koenderink, J., and Hoffman, D. Inferring Three-Dimensional Shapes from Two-Dimensional Silhouettes. *Journal of the Optical Society of America, A 4* (July 1987), 1168–1175.

[15] Saito, S., and Takahashi, T. Comprehensible Rendering of 3-D Shapes. *Computer Graphics 24*, 4 (August 1990), 197–206.